Backpropagation

Objetivo: Entender cómo aprenden las redes neuronales.

TAE IA
Formación de Talento Altamente Especializado

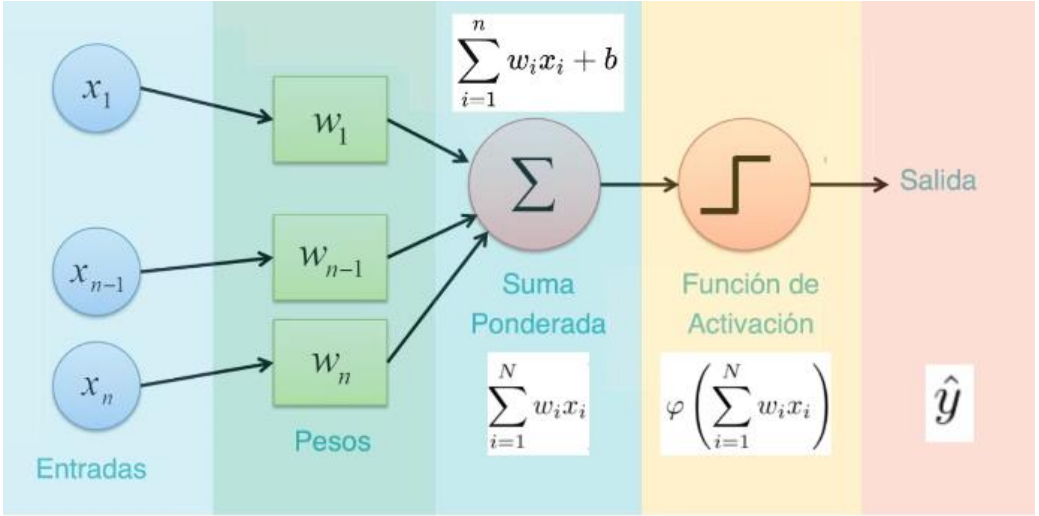
Introducción

Objetivo: Conocer el funcionamiento de las redes neuronales artificiales

Perceptrón

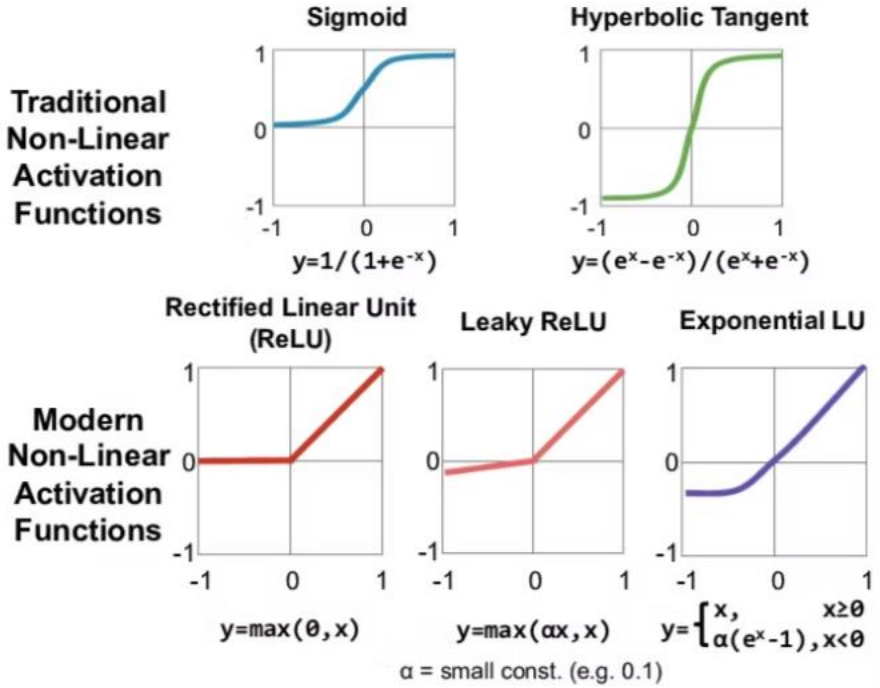
Estructura del Perceptrón:

- **Entrada:** Denotan los valores del perceptrón de características y la ocurrencia total de las características.
- **Pesos:** Cada entrada tiene un peso asociado que amplifica o atenúa la señal.
- **Suma ponderada:** Suma las señales de entrada.
- **Función de activación:** Aplica una función para determinar si la señal total es suficiente para activar la salida.
- **Salida:** Es la señal o valor final emitida por el perceptrón.



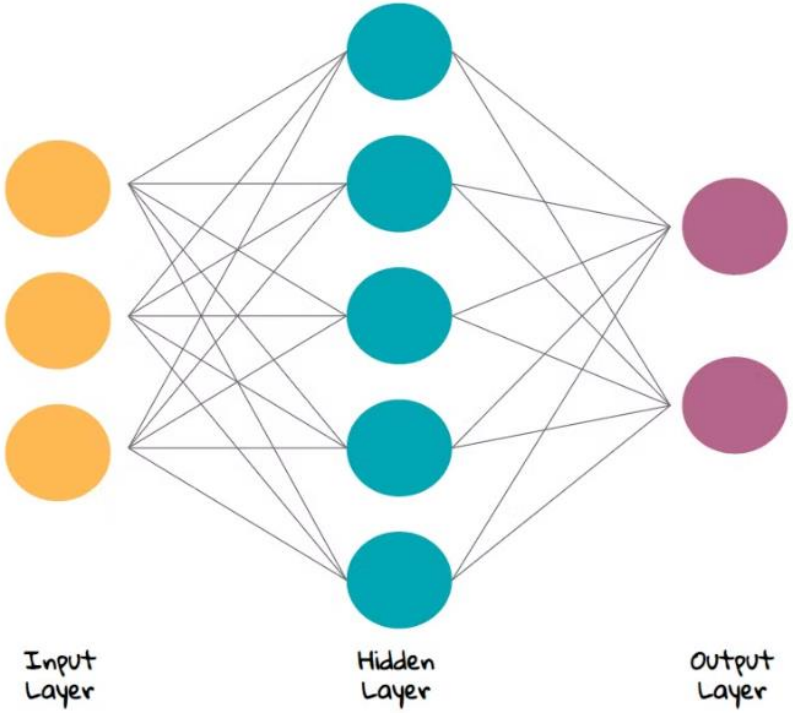
Modelo biológico de una neurona.

Función de activación



Fuente: https://ignaciogavilan.com/catalogo-de-componentes-de-redes-neuronales-ii-funciones-de-activacion/

Neural Network

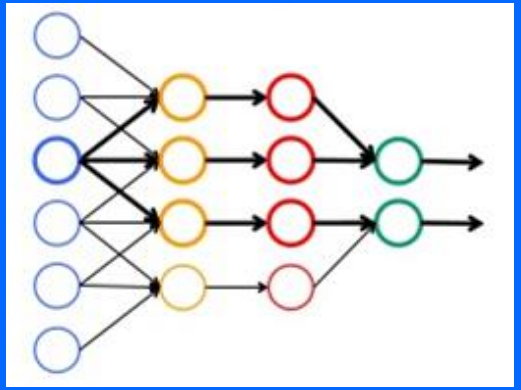


Backpropagation

Objetivo: Entender las dos fases de propagación de una red neuronal

Feedforward propagation

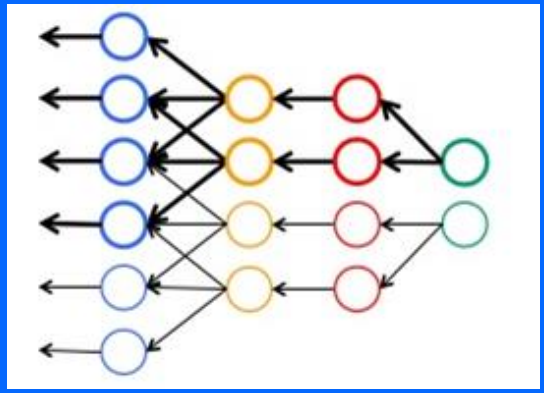
1. La red neuronal recibe una entrada como nuevo estímulo.
2. El estímulo se propaga desde la primera capa hasta la capa de salida.
3. La red genera una salida estimada.
4. La salida estimada se compara con la salida deseada y se calcula el error entre ambas.



TAE IA
Formación de Talento Altamente Especializado

Backward propagation

1. El error se propagan hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de la capa oculta.
2. Las neuronas de las capas ocultas reciben el error y se organizan a sí mismas para minimizar futuros errores.
3. Por cada señal de error modifican cada uno de sus valores libres w y b .



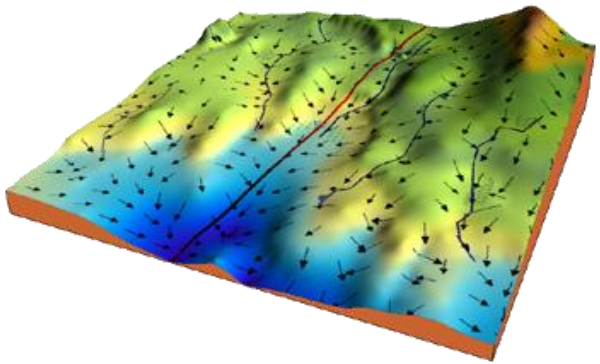
TAE IA
Formación de Talento Altamente Especializado

Gradient Descent

Objetivo: Entender cómo y porqué funciona el algoritmo gradiente descendiente.

Definición

- Es un algoritmo de optimización **iterativo** utilizado para encontrar un **mínimo local** de una **función diferenciable** moviéndose iterativamente en la dirección de descenso según lo definido por el **negativo del gradiente**.

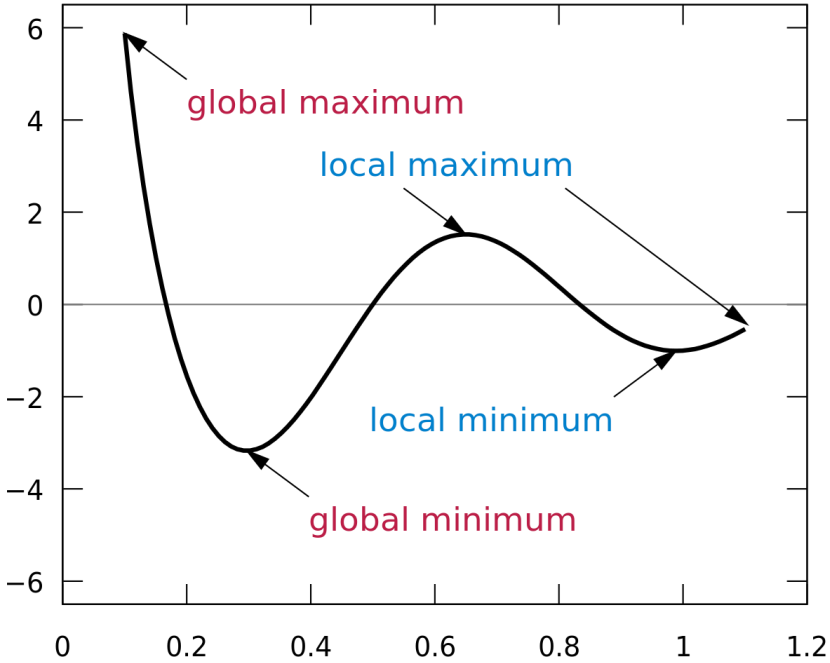


Fuente: link

TAE IA
Formación de Talento Altamente Especializado

Minimos y maximos

- Nuestra **funciones** de **costos** son de **error** por lo que buscaremos los **mínimos locales y globales,** es decir el mínimo error posible.
- Para **problemas complejos** es **difícil** encontrar los **mínimos globales.**



TAE IA
Formación de Talento Altamente Especializado

Función de pérdida

- La **función de pérdida** (Loss Function) cuantifica la cantidad en que la predicción se desvía de los valores deseados, es decir el **error durante el entrenamiento**.
- Nos indican qué tan **"bien"** es nuestro modelo para hacer **predicciones** para un **conjunto dado de parámetros**.
- Se denota con la siguiente función:

$$L(\hat{y}, y)$$

Gradiente de una función

- El "**gradiente**" es una **generalización multivariable de la derivada.**
- Es una **función** de **valor vectorial**, a **diferencia** de una **derivada**, que es una función de **valor escalar**.
- Los **valores** del **gradiente** nos dice cómo **actualizar** nuestros **parámetros** para hacer que el modelo sea más preciso.
- El **gradiente** para nuestra **función de pérdida** puede ver de la siguiente manera:

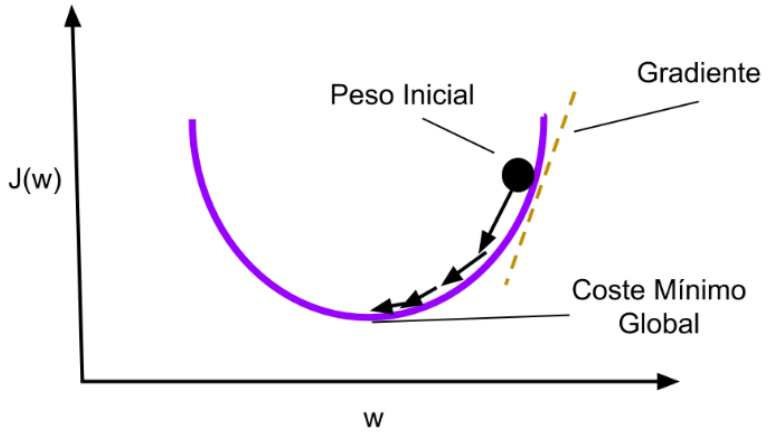
$$\nabla L(\hat{y}, y) = \left(\frac{\partial L(\hat{y}, y)}{\partial w_1}, \ldots, \frac{\partial L(\hat{y}, y)}{\partial w_n} \right)$$

Gradiente de una función - Derivada

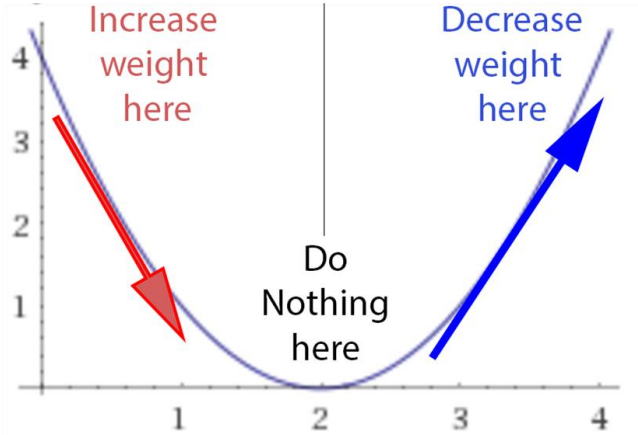
- La **derivada de la función**, mide la **rapidez** con la que **cambia** el valor de dicha función matemática según cambie el valor de su **variable independiente**.
- Al aplicar la **derivada a la función de pérdida** respecto a algún peso w o bias b, obtendremos un coeficiente de qué tanto se deben de **modificar** dicho **parámetro** para r**educir el error**.

Fuente: Wikipedia

Gradiente Descendiente



Fuente: Link

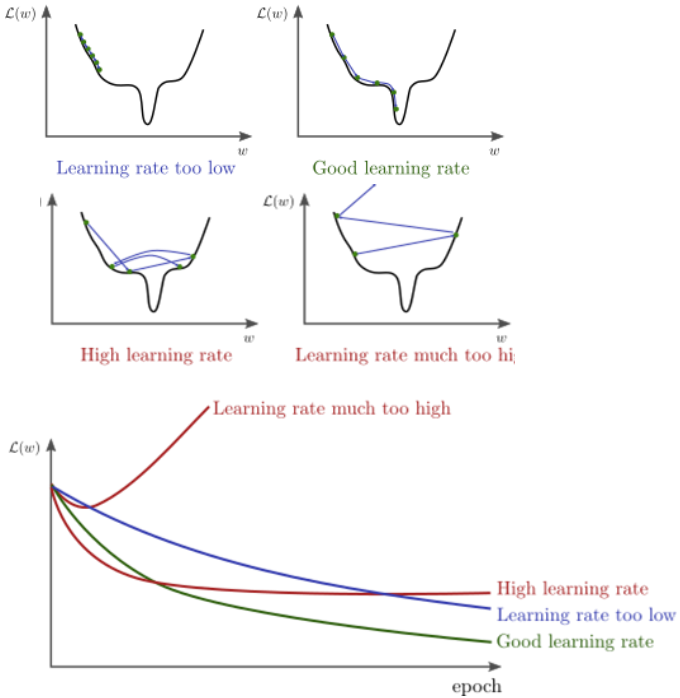


Fuente: Link

TAE IA
Formación de Talento Altamente Especializado

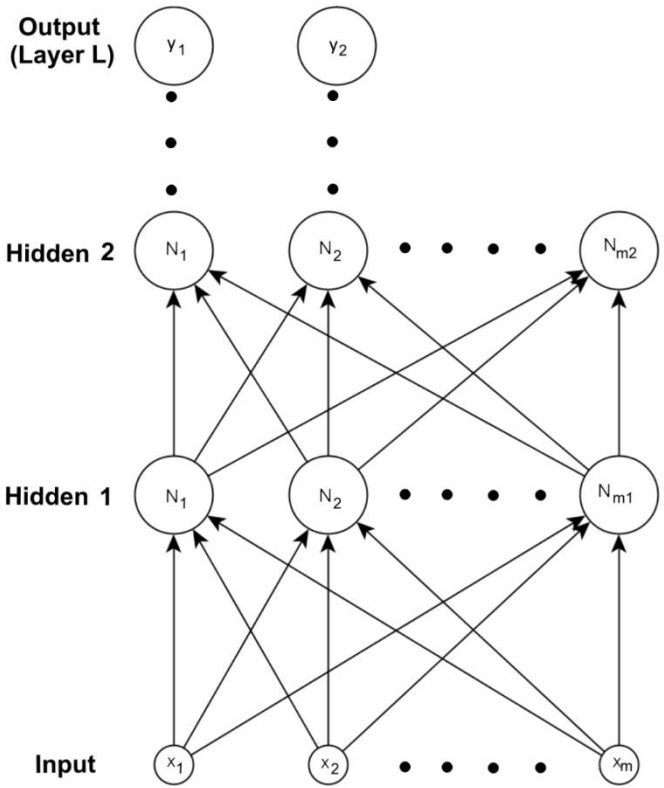
Tasa de aprendizaje (Learning rate)

- **Controla** la **rapidez o lentitud** con que un modelo de red neuronal **modifica sus parámetros** (aprende del problema).
- Con un **valor alto**, **aprender más** en cada paso, pero **corremos el riesgo** de sobrepasar el punto más bajo y que el **algoritmo nunca converga**.
- Con un **valor bajo**, **aprender más lento** en cada paso, por lo que tarda más tiempo en converger. Puede caer fácilmente en un mínimo local.



Fuente: https://cs231n.github.io/neural-networks-3/

TAE IA
Formación de Talento Altamente Especializado



Capa de salida

Se usa para hacer la predicción $\hat{y}$. Representamos la capa de salida como una regresión lineal (Regresión) o un regresión logística (Clasificación).

$$\hat{y} = W^l \mathbf{h}^{l-1} + b^l$$

Capa oculta

Las capas ocultas realizan una operación no lineal utilizando la salida de la capa anterior.

$$\mathbf{h}^k = \varphi(W^k \mathbf{h}^{k-1} + b^k)$$

Capa de entrada

Estos son los datos visibles o el estímulo de la red.

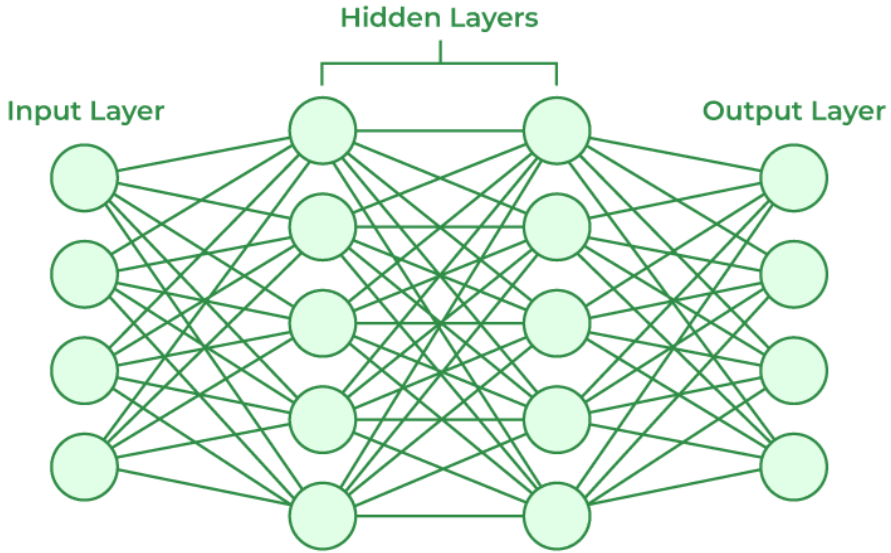
$$\mathbf{h}^0 = \mathbf{x}$$

TAE IA
Formación de Talento Altamente Especializado

Ejemplo: Red Neuronal de 2 capas ocultas

- **Capa de entrada**: $h^0 = x$
- **Primera Capa oculta:** $h^1 = \varphi(W^1 h^0 + b^1)$
- **Segunda Capa oculta:** $h^2 = \varphi(W^2 h^1 + b^2)$
- **Capa de salida:** $\hat{y} = W^3 h^2 + b^3$
- Donde:
  - x son los datos de entrada.
  - W^1, W^2, W^3 son las matrices de pesos para cada capa.
  - b^1, b^2, b^3 son los sesgos. Estos son valores numéricos.
  - φ es la función de activación no lineal en las capas ocultas.

Ejemplo: Red Neuronal de 2 capas ocultas



$$\hat{y} = W^3\varphi(W^2\varphi(W^1x + b^1) + b^2) + b^3$$

TAE IA
Formación de Talento Altamente Especializado

Fase de aprendizaje

- Podemos representar nuestro modelo de Machine Learning de la siguiente manera:

$$\hat{y} = f(x : \theta)$$

Donde:

- $f(x : \theta)$ -> **Representa** a nuestro **modelo de Red Neuronal.**
- θ -> Son los **parámetros** libres de la red, pesos W y bias b.
- $\hat{y}$ -> **Valor estimado** por la Red Neuronal.

Fase de aprendizaje - Función de pérdida

- La **función de pérdida** cuantifica la cantidad en que la predicción se desvía de los valores deseados, es decir el **error durante el entrenamiento**.

$$L(\hat{y}, y) = L(f(x : \theta), y)$$

Donde:

- $L()$ -> Funcion de perdida.
- y -> Valor real o esperado.
- $\hat{y}$ -> Valor estimado por la Red Neuronal.

Fase de aprendizaje - Actualización de parámetros

- Los parámetros de una función se actualizan de la siguiente manera:

$$\theta = \theta - \eta \nabla_\theta L(\hat{y}, y)$$

Donde:

- ∇_θ -> Gradiente respecto a los parámetros libres.
- η -> Tasa de aprendizaje. El valor va 0 a 1.

Fase de aprendizaje - Algoritmo

Algorithm 1 Gradient Descend

Input: Training set (X_{train}, Y_{train}), initial parameters θ, learning rate η and batch size m

Output: Trained parameters θ

1: **while** stopping criterion not satisfied **do**

2: Get a mini-batch of m examples from the training set $\{x^{(1)}, ..., x^{(m)}\}$ with corresponding targets $y^{(i)}$.

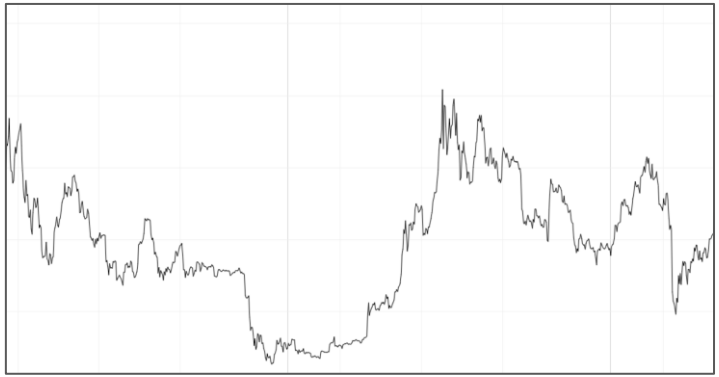
3: Compute gradient of loss function:
$\hat{\theta} \leftarrow \frac{1}{m} \sum_{i=0}^{m} \nabla_\theta L(f(x^{(i)} : \theta), y^{(i)})$

4: Update parameters:
$\theta \leftarrow \theta - \eta \cdot \hat{\theta}$

5: **return** θ

Actividad

- ¿Qué problemas tiene el algoritmo de gradiente descendiente?
- ¿Funcionaria bien para una función de pérdida como la siguiente?



- ¿Cómo mejorar se puede mejorar?

Optimización estocástica

- En estos **métodos** de **optimización** generan y usan **variables aleatorias**.
- Para **problemas estocásticos**, las **variables aleatorias** aparecen en la formulación del problema de optimización como **funciones aleatorias** o **restricciones aleatorias**.
- Los métodos de **optimización estocástica** también incluyen **métodos** con **iteraciones aleatorias**.

Variantes del Gradiente Descendiente

- **Stochastic gradient descent (SGD):** Gradiente descendiente con variables estocásticas.
- **Adagrad:** Ajusta la tasa de aprendizaje para cada parámetro individualmente, disminuyendo progresivamente a medida que se acumulan actualizaciones.
- **Adadelta:** Modifica Adagrad al limitar la acumulación de los gradientes, usando una ventana de tiempo móvil. No requiere una tasa de aprendizaje inicial explícita.

Variantes del Gradiente Descendiente

- **RMSprop:** RMSprop es un algoritmo de optimización que ajusta la tasa de aprendizaje para cada parámetro de forma adaptativa.
- **Adam:** Adam es una combinación de RMSprop y SGD con momentum. Mantiene dos promedios en movimiento: uno para los gradientes (como el momentum) y otro para los cuadrados de los gradientes (como RMSprop).
- **AdaMax:** Una variante de Adam que utiliza la norma L∞ en lugar de L2 para la actualización de los parámetros.

Regularización

Objetivo: Conocer qué es la regularización y algunos regularizadores.

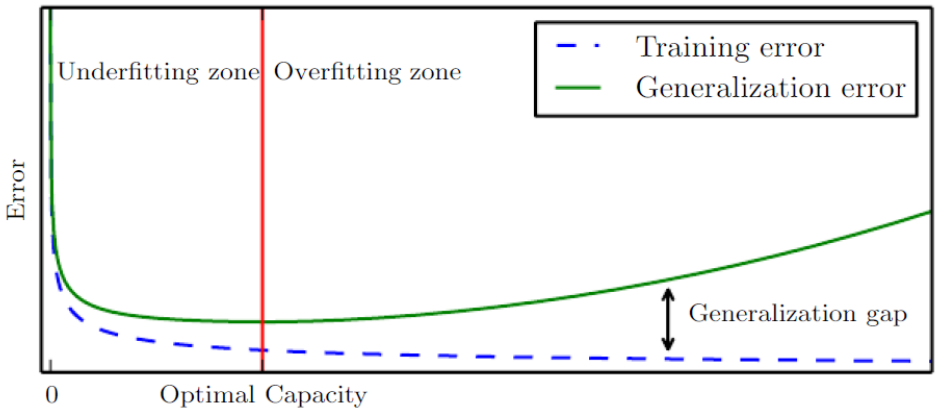
Regularización

- Un **problema central** en el **aprendizaje automático** es cómo hacer un algoritmo **pueda converger** y además que **funcione bien** no solo en los datos de **entrenamiento**, sino también en las **nuevas entradas**, (generalización).

- **Yoshua Bengio (2016)** definió la regularización como:

  "*Cualquier **modificación** que hagamos a un **algoritmo de aprendizaje** que tiene la intención de **reducir** su **error de generalización** pero no su error de entrenamiento*".

Regularización



Fuente: https://www.deeplearningbook.org/

Norma L2 (Weight decay, L2 Norm, Ridge)

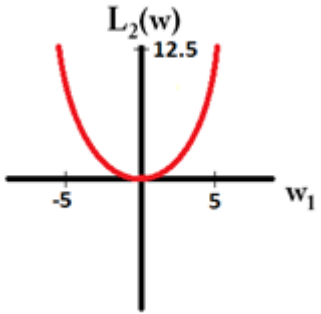
- La **penalización L2** se conoce comúnmente como **“weight decay”**.
- Esta estrategia de regularización lleva los parámetros libres cerca del origen (cerca al valor cero).
- **Penaliza** los parámetros con **valores grandes**.
- La **norma L2** agrega un término extra a la función pérdida:

$$L_2(\hat{y}, y) = L(\hat{y}, y) + \lambda_2 ||\theta||_2$$

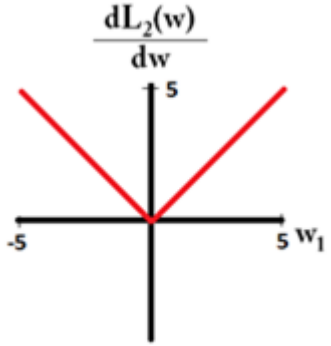
$$\|\boldsymbol{\theta}\|_2 := \sqrt{\theta_1^2 + \cdots + \theta_n^2}.$$

TAE IA
Formación de Talento Altamente Especializado

Norma L2 (Weight decay, L2 Norm, Ridge)



Función de la norma L2



Derivada de la norma L2

Observen la derivada de la norma L2

- ¿Por qué provoca que los parámetros w y b no crezcan demasiado?
- ¿Por qué es malo que los parámetros tienen valores muy grandes?

TAE IA
Formación de Talento Altamente Especializado

Norma L1 (L1 Norm, Lasso)

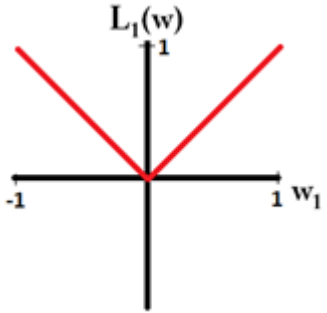
- El **objetivo** de la **norma L1** es **alcanzar modelos dispersos**.
- Eso significa que **moverá** ciertos **parámetros** hasta **cero**. Por lo que el modelo **remueve** las **características** menos importantes. **Se vuelve un selector de características importantes.**
- La **norma L1** agrega un término extra a la función pérdida:

$$L_1(\hat{y}, y) = L(\hat{y}, y) + \lambda_1 ||\theta||_1$$

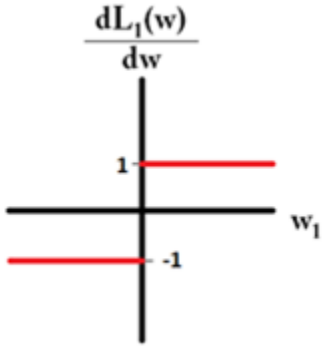
$$\|\boldsymbol{\theta}\|_1 := |\theta_1| + \cdots + |\theta_n|$$

TAE IA
Formación de Talento Altamente Especializado

Norma L1 (L1 Norm, Lasso)



Función de la norma L1



Derivada de la norma L1

Observen la derivada de la norma L1

- ¿Por qué provoca que los parámetros w y b lleguen a valor cero?
- ¿Por qué sólo remueve las características menos importantes?
- ¿Que representa en una red neuronal que un peso valga cero?

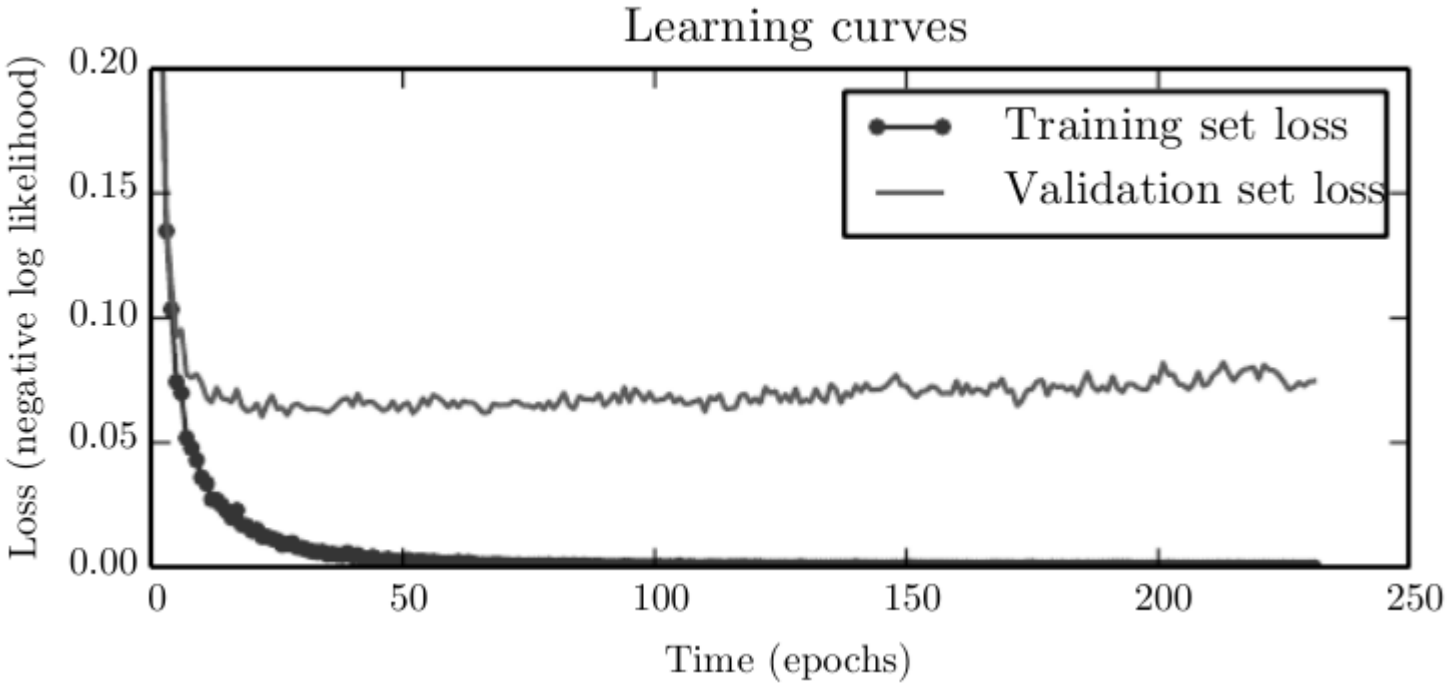
TAE IA
Formación de Talento Altamente Especializado

Overfitting y problema de parada de un modelo

- Al **entrenar** modelos **grandes** con **capacidad suficiente** para hacer **overfitting**, a menudo observamos que el **error de entrenamiento disminuye** constantemente con el tiempo, pero el **error del conjunto** de validación comienza a **aumentar nuevamente.**

- Enfoques:
  - Early Stopping.
  - Model Checkpoint

Overfitting y problema de parada de un modelo



Early stopping

- Es una **técnica** que **detiene** el **entrenamiento** cuando el rendimiento en el conjunto de validación deja de **mejorar**, para prevenir el overfitting.
- Usa un criterio de "**paciencia**" (número de épocas sin mejora) para **decidir cuándo detenerse**.
- El **objetivo** principal es r**educir el tiempo** de **entrenamiento** y prevenir el **overfitting**.

Model Checkpoint

- Es una **técnica** que guarda la **configuración de parámetros** del modelo durante el entrenamiento.
- Cuando **mejora** el error en el **conjunto de validación**, **almacenamos** una copia de los **parámetros** del **modelo**.
- Cuando **finaliza** el algoritmo de **entrenamiento**, devolvemos el **mejor modelo**.

Otros regularizadores

- Adaptive Learning Rate.
- Cross-Validation.
- Data Augmentation.
- Batch Normalization.
- Dropouts.
- Momentum.

Conclusión

Conclusiones

- La base de las redes neuronales es el perceptrón, de ahí se han generado diferentes arquitecturas que atacan problemas de manera específica.
- Las redes neuronales cuentan con tres tipos capas básicas, la capa de entrada, la capa oculta y la capa de salida.
- Una red neuronal puede tener cero o más capas ocultas, las cuales pueden contener la cantidad de neuronas que se requieran.
- Entre **más neuronas** tenga una red neuronal, es **más expresiva** y por lo tanto más compleja y **difícil de entrenar**.

Conclusiones

- Una **red neuronal aprende** gracias al algoritmo de **backpropagation**.
- Los **regularizadores** son **claves** para poder **entrenar** redes neuronales de **alta complejidad**.
- Para evitar entrenar tu red neuronal durante días, semanas o incluso indefinidamente, puedes aplicar la técnica de early stopping o limitar la cantidad de épocas de entrenamiento.

Bibliografía

- https://www.deeplearningbook.org
- https://cs231n.github.io/neural-networks-3/
- https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10
- https://medium.com/coinmonks/backpropagation-concept-explained-in-5-levels-of-difficulty-8b220a939db5
- https://medium.com/ai-quest/regularization-techniques-a-list-down-70df0e8693c
- https://medium.com/datathings/neural-networks-and-backpropagation-explained-in-a-simple-way-f540a3611f5e
- https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464

Código

Colaboratory

https://colab.research.google.com/drive/1wRIZPGgTzddSrE_-aRb6mNf44xoBMTly?usp=sharing

Gracias por su atención ¿Dudas?

TAE IA
Formación de Talento Altamente Especializado